In the claims

1. (currently amended) A method comprising:

executing a code section relating to memory using a hardware transactional approach to executing code sections, ~~the hardware transactional approach to executing code sections implemented in hardware other than a processor~~ where the hardware transactional approach to executing code sections is employed to attempt to execute the code section before a software approach to locking memory is ever employed to attempt to execute the code section, such that a first time the code section is attempted to be executed the software approach to locking memory is not employed to attempt to execute the code section prior to the hardware transactional approach being employed to attempt to execute the code section; and,

where the hardware transactional approach executing code sections fails a threshold in executing the code section, executing the code section relating to the memory using a software approach to locking memory.

2. (previously presented) The method of claim 1, further comprising executing other code sections relating to the memory using the software approach to locking memory.

3. (previously presented) The method of claim 1, wherein executing the code section relating to the memory using the hardware transactional approach to executing code sections comprises:

starting a transaction inclusive of the code section; and,

conditionally executing the transaction.

4. (previously presented) The method of claim 3, wherein executing the code section relating to the memory using the hardware transactional approach to executing code sections

further comprises, upon successfully completing the transaction, committing execution of the transaction to the memory to which the code section relates.

5. (previously presented) The method of claim 1, wherein the hardware transactional approach to executing code sections failing the threshold in executing the code section comprises the hardware transactional approach to executing code sections aborting execution of the code section a single time.

6. (previously presented) The method of claim 1, wherein the hardware transactional approach to executing code sections failing the threshold in executing the code section comprises the hardware transactional approach to executing code sections aborting execution of the code section a predetermined plurality of times.

7. (previously presented) The method of claim 1, wherein the hardware transactional approach to executing code sections failing the threshold in executing the code section comprises utilizing a digital filter as the threshold in determining whether to utilize the software approach to locking memory to execute the code section.

8. (previously presented) The method of claim 1, wherein the hardware transactional approach to executing code sections failing the threshold in executing the code section comprises utilizing information passed from a compiler to determine whether to utilize the software approach to locking memory to execute the code section.

9. (previously presented) The method of claim 1, wherein the hardware transactional approach to executing code sections failing the threshold in executing the code section comprises tracking a success rate of the hardware transactional approach to executing code sections in

executing the code section to determine whether to utilize the software approach to locking memory to execute the code section.

10. (previously presented) The method of claim 1, wherein executing the code section relating to the memory using the software approach to locking memory comprises:

placing a lock on the memory to which the code section relates;

executing the code section;

committing execution of the code section to the memory as the code section is executed; and,

removing the lock on the memory to which the code section relates.

11. (previously presented) The method of claim 1, wherein executing the code section relating to the memory using the software approach to locking memory comprises utilizing one of a queued lock, a non-uniform memory architecture (NUMA) lock, and a reader-writer lock.

12.-20. (cancelled)